HDF5 Issues raised by VORPAL

Presenter: J.R. Cary[†*]
[†]Tech-X Corporation; [*]University of Colorado
in collaboration with Cameron Geddes, P. Messmer

January 20, 2009

- What does VORPAL do?
- What kinds of data does it output? (fields, particles, geometries)
- What performance are we seeing?
- Other problem areas (h5diff)
- Priorities

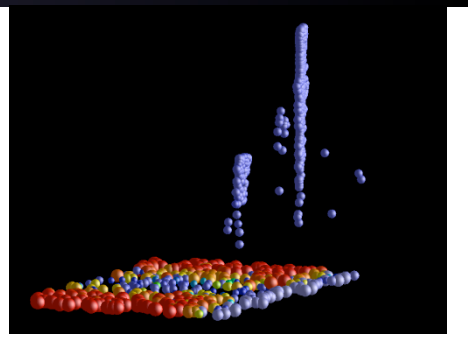
Acknowledgments

- Thanks to the VORPAL Team: T. Austin, G.I. Bell, D.L. Bruhwiler, R.S. Busby, J. Carlsson, J.R. Cary, B.M. Cowan, D.A. Dimitrov, A. Hakim, J. Loverich, P. Messmer, P.J. Mullowney, C. Nieter, K. Paul, S.W. Sides, N.D. Sizemore, D.N. Smithe, P.H. Stoltz, S.A. Veitzer, D.J. Wade-Stein, G.R. Werner, M. Wrobel, N. Xiang, W. Ye
- Thanks to the VizSchema Team: T. Austin, A. Hakim, J. R. Cary, S. Veitzer, A. Pletzer, D.N. Smithe, M. Miah, P. Stoltz, S. Shasharina, P. Hamill, S. Kruger, D. Alexanda, P. Messmer
- Thanks to the FACETS Team:A. Pletzer, A. Hakim, M. Miah, S.E. Kruger, S. Vadlamani, J. Carlsson, J.R. Cary, A. Pankin, R. Cohen, T. Rognlien, T. Epperly, D. McCune, K. Indireshkumar, G. Hammett, A. Pigarov, A.D. Malony, A. Morris, S. Shende, L.C. McInnes, H. Zhang, J. Larson, D. Estep, J. Cobb
- Thanks for support: DOE HEP, FES, NP, SciDAC, SBIR; NSF; AFOSR, JTO, DOD SBIR, NASA

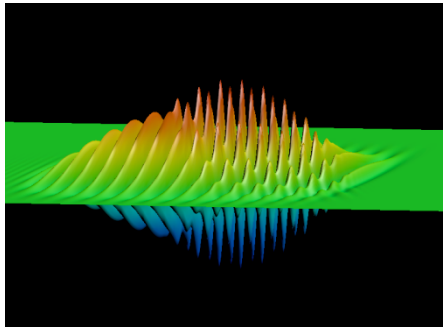
Our implementations are in the VORPAL Computational Framework

- Laser-plasma and laser-gas interactions (collab. with LBNL)
  - multiple invited talks at DPP, PAC
  - PRL's, Nature cover, …
- Electron cooling for RHIC (collab. with Brookhaven National Lab)
- Thruster modeling (DOD)
- Electromagnetic cavities
- Photonic Band Gap structures
- Recognized as one of the SciDAC codes
- Originally supported by NSF, but most of the subsequent development supported by HEP-TECH, NP, OFES, AFOSR

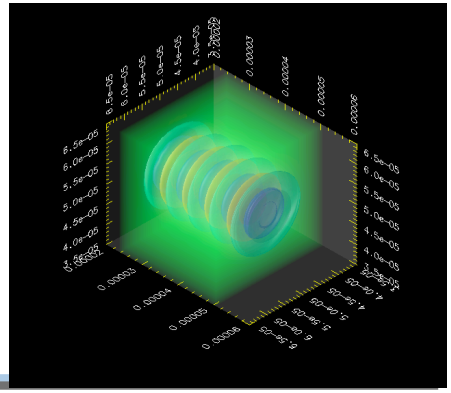
We layer on the C libs



Particle beams



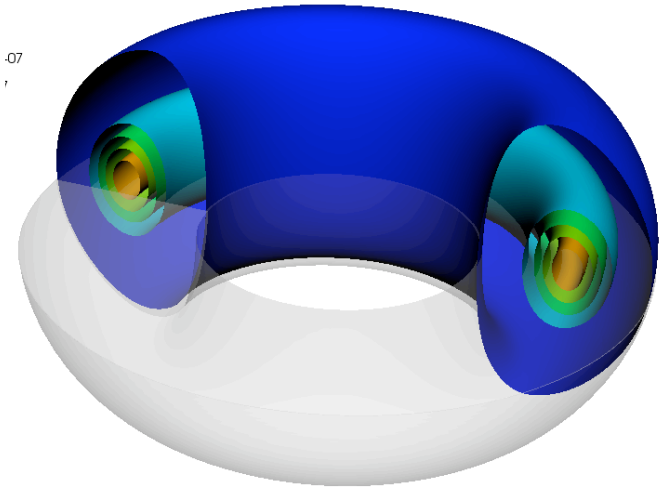
Colliding laser pulses

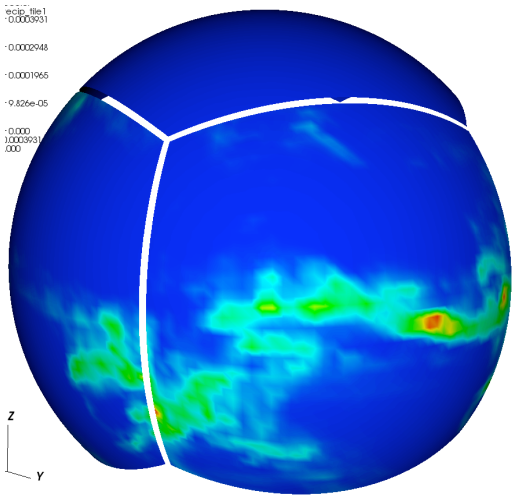


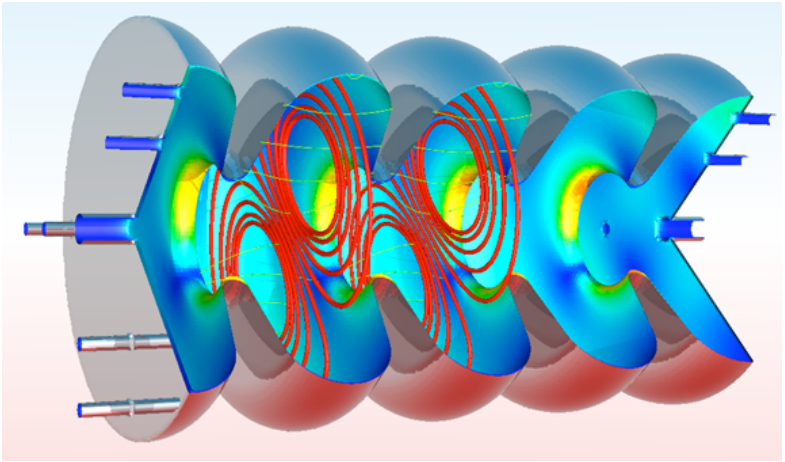
Wake fields

But our use goes far beyond to many applications

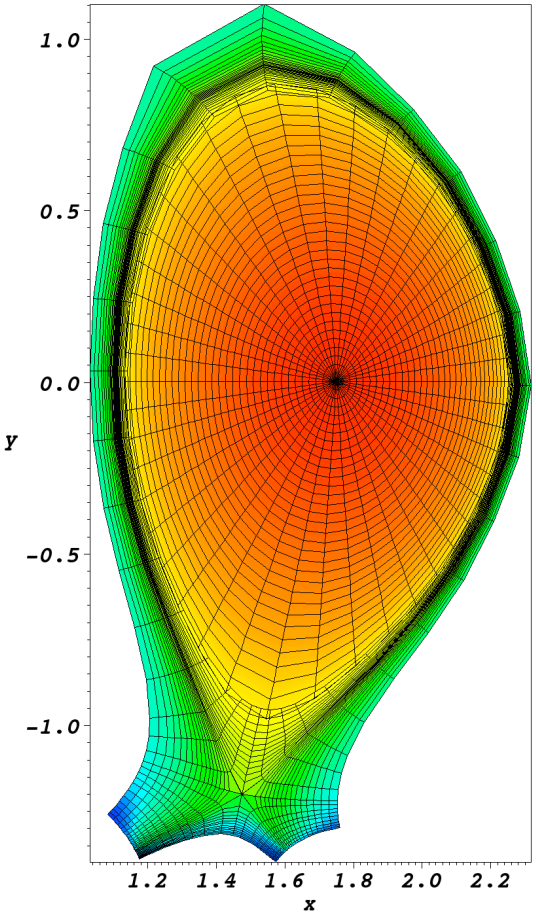
https://ice.txcorp.com/trac/vizschema







fusion
accelerators
climate

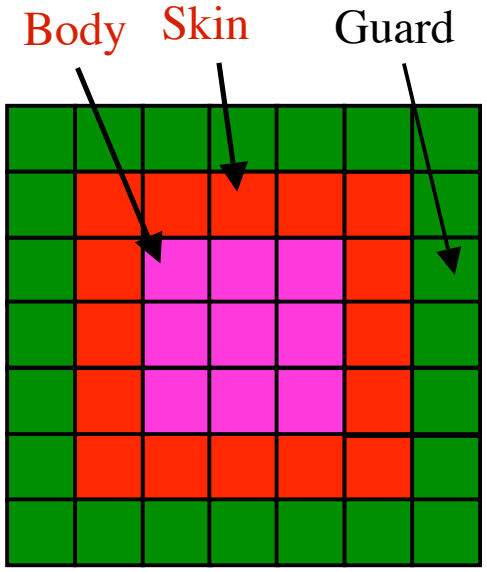


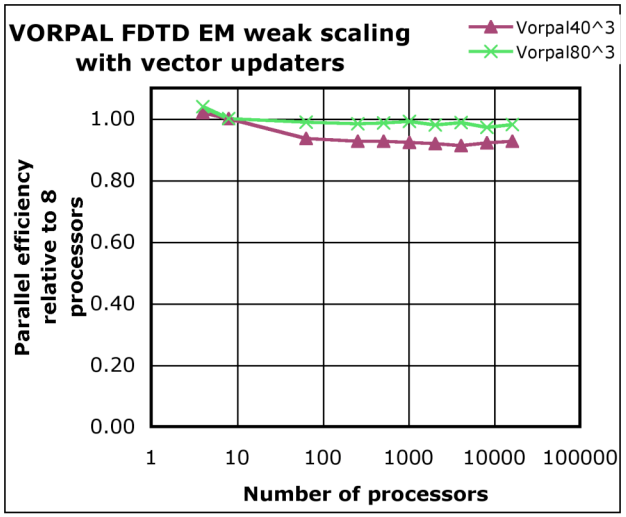
VORPAL outputs (mainly) two kinds of data

- **Particle data:**
  - Each processor outputs into own segment of file
  - Restarting on a different number of procs is a killer
- **Field data:**
  - Rank 4, strided
  - Component minor, C: [ix][iy][iz][ic]

Finite-difference time-domain computations very efficient

- All communications through boundary
- Measure is scaling
  - Weak: region size per processor constant
  - Strong: total region remains of constant size

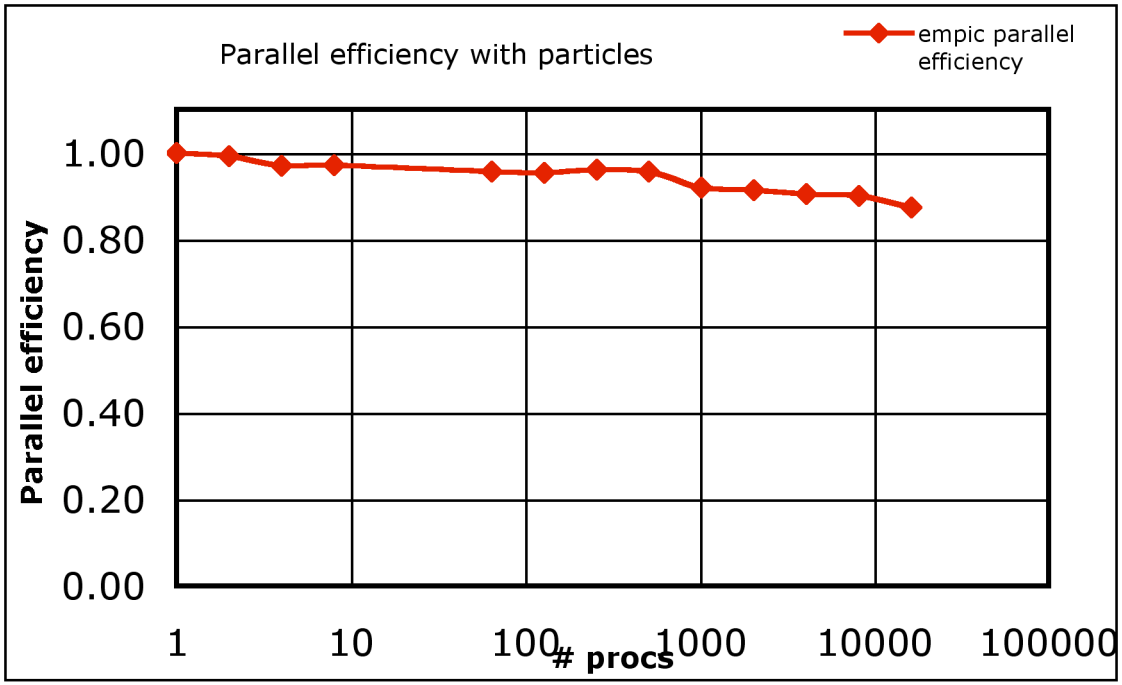




Lastest GPU speedup: 45x

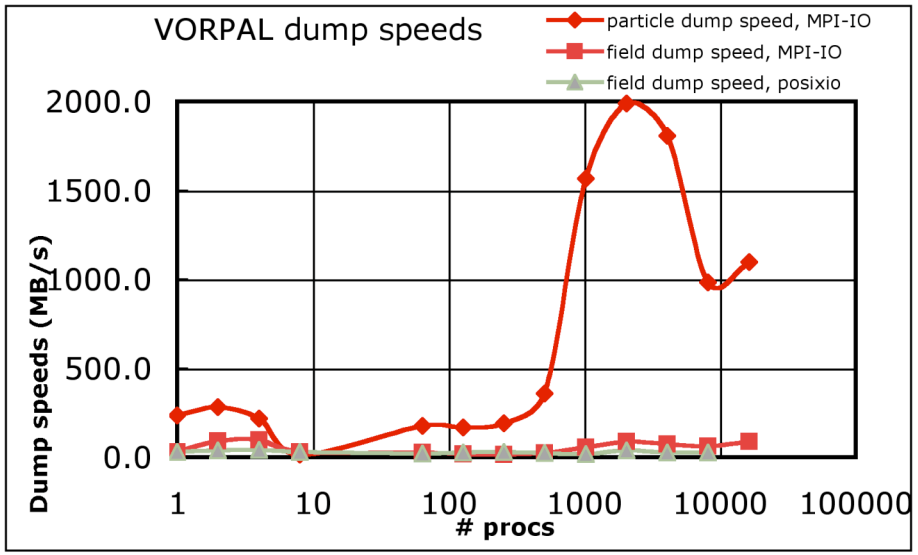
Need I/O that works well with 1.5 MB per proc

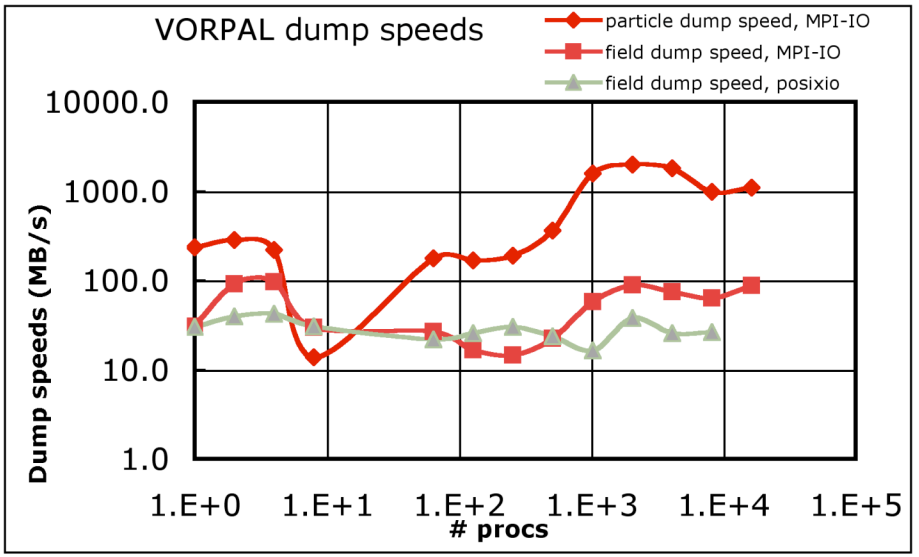
High efficiency with particles as well



- 40^3 domains
- 90% efficiency at 16k procs
- Many ideas for improving

But dumping (except for the bump) suffers from nonscaling (no speedup?)





No accounting for I/O storms

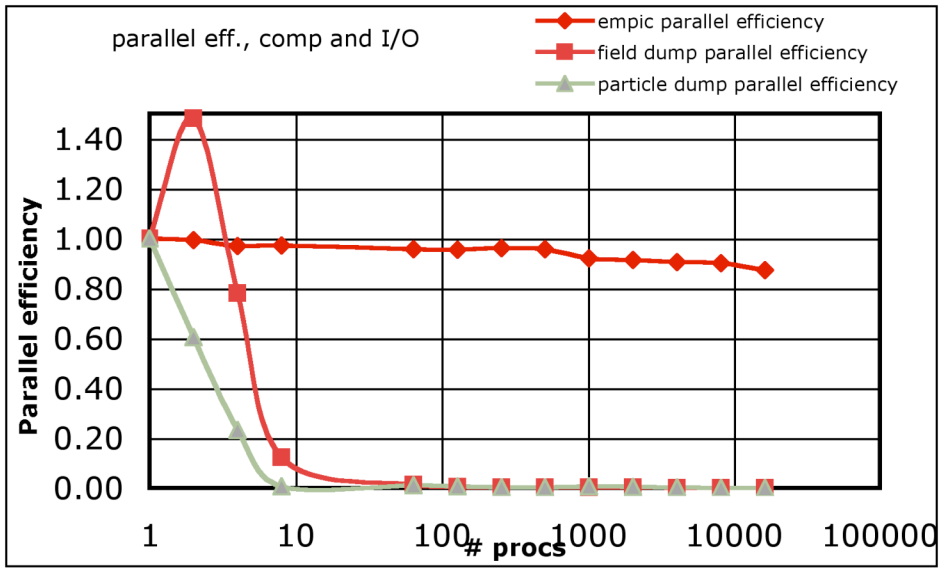
- **We have chased multiple tales**
- **"Don't use attributes"**
- **Use Posix IO**
- **Do chunking**

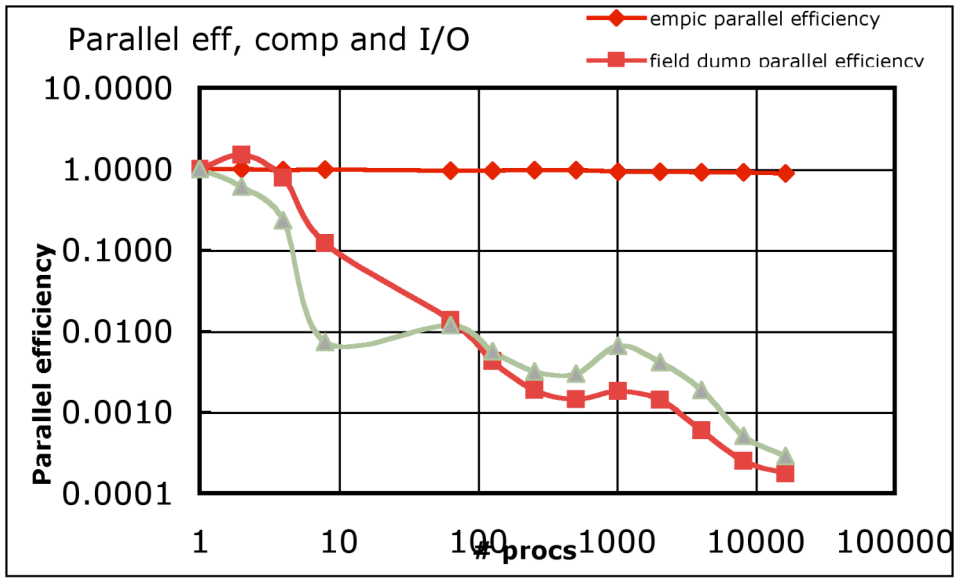
  None of these helped much, some hurt

It was even worse for uneven domain sizes

- Users pick an over problem size (e.g., 2750x150x150)
- VORPAL uses a prime factorization of the number of procs to determine domain size
  - 78 procs = 13x3x2
  - (2750/13)x150x150 = (211±1)x150x150
  - ((211±1)/3)x150x150 = (70±1)x150x150
  - (70±1)x(150/2)x150 = (70±1)x75x150
- Observed 10x slowdown or more on 1000's of procs with uneven domains

Really looks bad on a parallel efficiency plot: 4 orders of magnitude bad



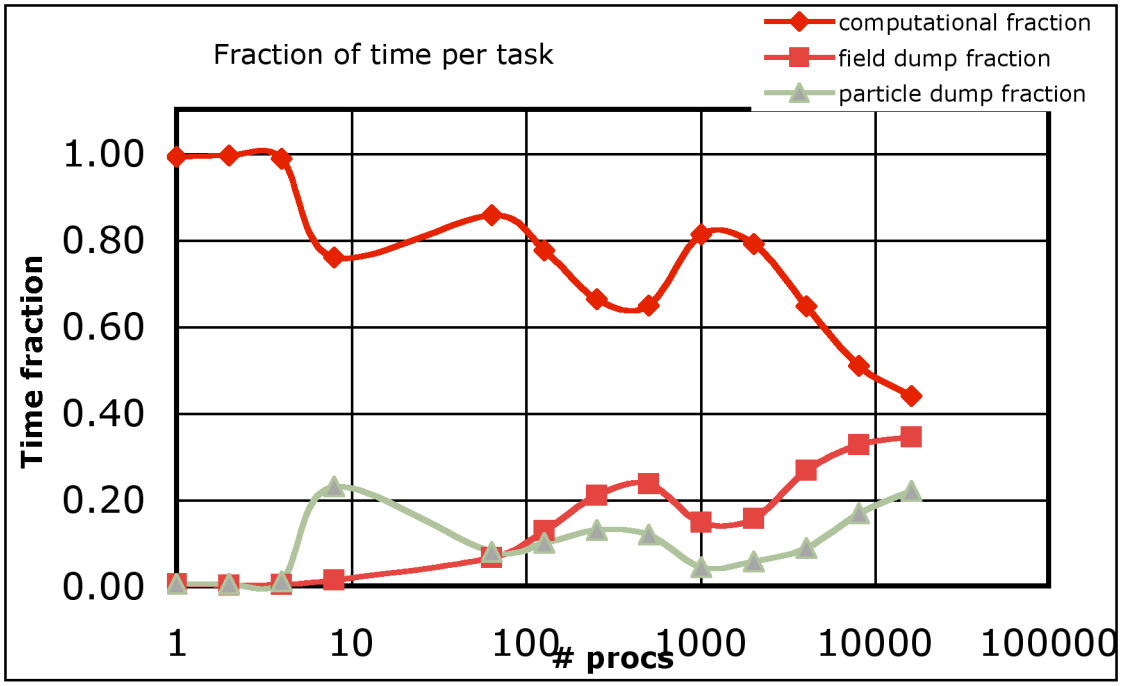


But not really this bad….

We have chosen a physics based I/O to computation benchmark

- **Electromagnetics**
  - Frequency extraction requires dumps of the order of a period = the time to cross the simulation
- **Electromagnetic particle in cell (EMPIC)**
  - Can be much longer
  - The results may overstate the importance of I/O in some cases
- **Nevertheless: For a 1000-cell (per direction) simulation, dump every 1000 steps.**

With this benchmark, computation is no longer dominant at 16k procs



For the first time, I/O is greater than 50% of time!

We rely on h5diff for our tests

- Run on multiple platforms
- Hundreds of tests
- Compare with previous results using h5diff
- Send out notifications

Problems with h5diff lead to need to wrap h5diff

- Desired capability
  - Specify fuzz by component
  - Return consistent error code

```
$ h5diff pseudoPotential1s_History-2.h5 pseudoPotential1s_History.h5
--------------------------------
Some objects are not comparable
--------------------------------
Use -v for a list of objects.
$ echo $?
0
```

- Crashes when attributes have different types
- So our wrapper
  - Uses h5dump to determine contents
  - Captures crashes, type differences
  - Result code differs depending on whether data or attributes differ

Also found dramatic slowdown in going to 1.8.1

```
[txqaauto@boron empic]$ time h5diff unideltaf3p_electrons_1.h5
  unideltaf3p_electrons_1.h5
real    0m39.337s
user    0m36.776s
sys     0m0.323s
[txqaauto@boron empic]$ time /contrib/hdf5-1.6.7/bin/h5diff
  unideltaf3p_electrons_1.h5 unideltaf3p_electrons_1.h5
real    0m0.601s
user    0m0.215s
sys     0m0.177s
```

Multiple tasks: priority order

- Strided data
  - otherwise I have to turn strided data into non strided -- build my own layer
- h5diff
  - Fix crashes
  - Add per component diffs
- Particle data is reaching the limits of IOTs?